

Be careful of buffer length when using returned value

Sean Barnum, Cigital, Inc. [vita¹]

Copyright © 2007 Cigital, Inc.

2007-03-23

Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 3125 bytes

Attack Category	<ul style="list-style-type: none">Malicious Input		
Vulnerability Category	<ul style="list-style-type: none">Buffer Overflow		
Software Context	<ul style="list-style-type: none">OtherSystem Configuration		
Location	<ul style="list-style-type: none">stdlib.h		
Description	<p>getenv() allows a programmer to get the value of an environment variable.</p> <p>The value of this variable is stored in a character array and the size of the return value is not known when the function is called. Use caution (memcpy(), strncpy()) when copying the returned buffer.</p>		
APIs	FunctionName	Comments	
	getenv()		
Method of Attack	If strcpy() or another vulnerable function is used on the return value of this function, an attacker could specify an arbitrarily long environment variable and overflow this buffer.		
ExceptionCriteria	If strncpy or memcpy is used with the appropriate length, this function is safe to use in tandem.		
Solutions	Solution Applicability	Solution Description	Solution Efficacy
	This solution is always applicable	Size any buffer that the environment variable will be copied into appropriately and copy in the data using strcpy() or memcpy() with the appropriate limit.	This solution is always effective.

1. <http://buildsecurityin.us-cert.gov/bsi-rules/35-BSI.html> (Barnum, Sean)

Signature Details	char * getenv(const char *name);	
Examples of Incorrect Code	<pre>/* Improperly sized buffer */ char path = getenv("PATH"); char target [10]; strcpy(target, path);</pre>	
Examples of Corrected Code	<pre>/* Properly-sized buffer */ char path = getenv("PATH"); int buf_len = 50; char target [buf_len]; strcpy(target, path, buf_len);</pre>	
SourceReferences	<ul style="list-style-type: none"> • getenv() man page² • ITS4 Source Code Vulnerability Scanning Tool³ • Rough Auditing Tool for Security (RATS)⁴ 	
Recommended Resource		
Discriminant Set	Operating Systems	<ul style="list-style-type: none"> • UNIX (All) • Windows (All)
	Languages	<ul style="list-style-type: none"> • C • C++

Cigital, Inc. Copyright

Copyright © Cigital, Inc. 2005-2007. Cigital retains copyrights to this material.

Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and “No Warranty” statements are included with all reproductions and derivative works.

For information regarding external or commercial use of copyrighted materials owned by Cigital, including information about “Fair Use,” contact Cigital at copyright@cigital.com¹.

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

1. <mailto:copyright@cigital.com>